

AD-A034 524

ROCK ISLAND ARSENAL ILL GENERAL THOMAS J RODMAN LAB
MEDIT - A PROGRAM TO EDIT COMPUTER SOURCE PROGRAMS. (U)

APR 76 J J HURT

UNCLASSIFIED

RIA-R-TR-76-010

F/G 9/2

NL

| OF |
AD
A034524



END

DATE
FILMED
2-77

ADA 034524



1
B.S.

AD

R-TR-76-010

MEDIT

A PROGRAM TO EDIT
COMPUTER SOURCE PROGRAMS

by

JAMES J. HURT

1 APRIL 1976



PREPARED BY

RESEARCH DIRECTORATE

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION

PREPARED FOR

RESEARCH DIRECTORATE
GENERAL THOMAS J. RODMAN LABORATORY
ROCK ISLAND ARSENAL
ROCK ISLAND, ILLINOIS 61201

DISCLAIMER

The findings of this report are not to be construed as an official department of the Army position, unless so designated by other authorized documents.

DISPOSITION INSTRUCTIONS

Destroy this report when no longer needed. Do not return to the originator.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER R-TR-76-010	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MEDIT - A Program to Edit Computer Source Programs		5. TYPE OF REPORT & PERIOD COVERED 9 Final rept. 09
7. AUTHOR(s) James J. Hurt		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Research Directorate, SARRI-LR-S GEN Thomas J. Rodman Laboratory Rock Island Arsenal, Rock Island, IL 61201		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 16 1T161101A91A
11. CONTROLLING OFFICE NAME AND ADDRESS DAMA-ARZ-B Chief of Research and Development HQ, Department of the Army		12. REPORT DATE 1 Apr 76
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES 27 0135p.
16. DISTRIBUTION STATEMENT (of this Report) APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED		15. SECURITY CLASS. (of this report) UNCLASSIFIED
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE JAN 18 1977
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Software Maintenance ADP Utilities		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A portable and easily used source text editor is described. This computer utility provides a method to maintain and modify computer source text in a manner that is very easy to use, provides a complete audit trail of changes, and can be used on a wide variety of computers, eg IBM-360-370, CDC-6000, UNIVAC-1100, et al.		

PREFACE

MEDIT is a computer program designed to assist a programmer in the maintenance of a large source library. Such a program is provided as an utility program by every major computer manufacturer, therefore some justification must be given for the subject of this report.

The author is familiar with the following vendor supplied utility programs:

- a. IEBUPDAT for IBM-360 and IBM-370 computers.
- b. IEBUPDTE for IBM-360 and IBM-370 computers.
- c. SETUP for CDC-6000 computers.
- d. UPDATE for CDC-6000 computers.
- e. EDIT, written by the Systems Research Group, The Ohio State University.

In addition, the author is cognizant of similar utility programs supplied by UNIVAC, Burroughs, Digital Equipment Corporation, and Computer Sciences Corporation. The following comparisons are based on the author's experience:

- i. EDIT is the easiest to use.
- ii. UPDATE provides the most useful features.
- iii. None of these are portable - none can be easily transferred to another vendor's hardware.

The intent in writing MEDIT was to provide a source library maintenance utility that was easy to use (as is EDIT), that provided useful capabilities (as UPDATE), and that can be easily transported from one machine to another. The actual program described must be viewed as Version 1 of the desired program. MEDIT is as easy to use as EDIT but does not provide all the useful features of UPDATE. Future versions of this program will include some of these missing features.



iii The following page is blank.

TABLE OF CONTENTS

	<u>PAGE</u>
1.0 INTRODUCTION	1-1
2.0 CREATING A BASE MODULE	2-1
3.0 CHANGING AND TESTING	3-1
4.0 CREATING A NEW BASE MODULE	4-1
A.0 APPENDIX - LISTING OF THE MEDIT PROGRAM	A-1

LIST OF FIGURES

<u>Figure</u>	<u>PAGE</u>
2.1 First Part of an Edit Command Deck to Create a Base Module .	2-2
2.2 First Page of MEDIT Output when BAsE Module is Created . . .	2-3
2.3 Job Control to Create a Base Module	2-4
3.1 Sample Edit Command Deck	3-2
3.2 MEDIT Output from Sample Edit Command Deck	3-3
3.3 Job Control to Edit a Module	3-6
4.1 Job Control to Update Base Module	4-2

1.0 INTRODUCTION

Basically, there are two techniques of developing a computer program. The first technique is to maintain a deck of the program's source statements and make changes to the program by manual insertion, removal, or replacement of cards in this deck. This manual technique is adequate if the number of cards in the deck is small, say less than 500. When the source deck becomes large, or several people are making simultaneous changes, this manual technique becomes inadequate: every computer run involves handling a large number of cards; no audit trail is kept of the changes that have been made; and locating the correct cards to change is difficult, time consuming, and prone to error.

The second technique is to use a computer program called an editor to make the desired changes. The source deck is stored on disk or tape so that large numbers of cards do not have to be handled. The source cards are numbered so that location of a certain card is easy and natural. The changes, and only the changes, are punched on cards. These cards serve as a record of the changes that have been made. These editor programs usually permit the separation of the entire source program into independent subunits (for example, a subroutine or function) and permit reference to one or more of these subunits without having to refer to the others. The complete source program is called a library and each of the subunits is called a module.

The computer program MEDIT is designed for the maintenance of a large library consisting of one or more modules. The use of MEDIT proceeds in several steps. The first step, described in Chapter 2, is to use MEDIT to create a base module of the original source decks. This base module consists of a tape or disk file with 80 characters per record (card image) and with sequence numbers in columns 77 through 80. The second step, described in Chapter 3, is to make and test changes to this base module. This consists of running MEDIT to create a test module consisting of the base module as altered by edit commands, then compiling and executing this test module. Once the changes have proven successful, they can be made permanent by creating a new base module as described in Chapter 4. Creating a new base module should be avoided if at all possible. The time to create a new base module is either:

- a. When production runs of the test version are nearly completed and the user is very sure that no additional changes will be required; or
- b. When the edit command deck exceeds 1000 cards, approximately one-half a box of cards.

The base module is a sequential file of card images with sequence numbers in columns 77 - 80. Columns 1 - 72 contain data or source statements. This file can be used as the source for a compiler. An MEDIT run produces a module called a test module that reflects every change to the base module and is suitable for compilation. Columns 73 - 76 are not used in the base module. These columns are used in the test module.

The input to MEDIT consists of control cards that give a detailed description of the changes to be made to the base module. A MEDIT run provides a listing of all the control cards and a detailed listing of all the changes made on this run. The first control card serves as a title that is printed at the top of every page of the listing.

The format for an edit command is:

Columns 1 - 2:

Two characters that identify this card as an edit command. While the two characters .. are used in this report, any other combination of two characters may be used. Examples are /. and ./ and \$\$ and **. Do not mix the characters in any one MEDIT run -- use the same characters throughout.

Column 3:

One character that identifies the type of MEDIT command. Chapter 3 lists the valid MEDIT commands and their meanings.

Columns 4 - k:

One or two decimal numbers. A comma separates the numbers if there are two numbers.

Columns k+1-72:

See a description of the E-Command. Placing comments here that describe the change being made is highly recommended.

Columns 73 - 80:

Not used by MEDIT and not listed.

Any card that is not a MEDIT command, i.e., any card whose first two characters disagree with columns 1 - 2 of an edit command, is treated as

a data card to be inserted into the test module. The placement of inserted cards is determined by the latest preceding edit command.

2.0 CREATING A BASE MODULE

Before a module can be updated, it must be created. The program MEDIT is written to make library creation easy. A module is created by reading the deck of cards in the desired order. Figure 2.1 shows a sample edit command deck to create a base module. Figure 2.2 shows the output from MEDIT that results from these edit command cards.

The base module is a copy of the control deck except that the cards in each deck are numbered sequentially in columns 77 - 80 for later reference. If this base module is then compiled, a complete listing of the base module together with sequence numbers will be obtained. This listing should be kept as a reference for as long as the base module is kept. The binary deck output from this compilation may also be saved.

In summary, the MEDIT run to create a library should result in:

- a. A base module with sequence numbers;
- b. A listing of the contents of the base module; and
- c. The compiled version of the base module.

The various parts of Figure 2.3 show the job control cards to create a base module without compilation at each installation where MEDIT has been implemented.

```

C.... MEDIT VERSION TWO MEDIT VERSION TWO MEDIT VERSION TWO
C.. UNIT USAGE ENVIRONMENT TYPE
C- 1 BASE (SOURCE TO BE EDITED) F(80) TAPE IN
C- 2 TEST (RESULT OF EDITING BASE) F(80) TAPE OUT
C 5 CMND (EDIT COMMAND CARDS) F(80) CARD IN
C 6 LIST (RESULTS OF EDIT) FA(85) PRINT

C... FIRST CMND CARD IS USED AS A TITLE
C.. (PRINTED AT TOP OF EVERY PAGE OF LIST)
C- FIRST TWO COLUMNS OF FIRST CMND CARD IDENTIFY EDIT COMMANDS
C ANY CMND CARD NOT A VALID EDIT COMMAND IS INSERTED IN TEST
C.... THIRD COLUMN OF EACH EDIT COMMAND CARD IS TYPE
C.. VALID EDIT TYPES ARE-
C.. +N INSERT AFTER CARD NUMBER N
C- -N DELETE CARD N
C -N,M DELETE CARDS N THROUGH M INCLUSIVE
C.... CN MAKE CARD N A COMMENT CARD
C.. CN,M MAKE CARDS N THROUGH M COMMENT CARDS
C.. UN UNCOMMENT CARD N
C- UN,M UNCOMMENT CARDS N THROUGH M INCLUSIVE
C ENXAXBX EDTT CARD N
C EN,MXAXBX EDIT CARDS N THROUGH M INCLUSIVE

C
C SWITCH THAT INDICATES ENDFILE ON BASE
LOGICAL#1 ENDSW
C
C DECLARE ALL OTHER VARYABLES TO BE TYPE INTEGER
INTEGER#2NTYPE,TYPE(5)
INTEGER#2CMND(72),TITLE(72),OLCD(72),MED(5)
INTEGER#2OLD,NEW,PAGE,LINE,MAXLIN,NERRS,CHAR(2),THIS,NCOL
INTEGER#2SPACE,COMMA,DIGIT(10)
INTEGER#2 NUM,I
C***** FOLLOWING LOGIC IS FOR TEXT EDIT FEATURE ONLY *****
C***** INTEGER#2 NEWCD(73),LOLD,POLD,LNEW,RNEW,ICOL,JCOL,KCOL
C***** PRECEDING LOGIC IS FOR TEXT EDIT FEATURE ONLY *****
C
C DATA INITIALIZATION
DATA SPACE/1H/,COMMA/1H/,MED/1HM,1HE,1HD,1HI,1HT,1H/
DATA DIGIT/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
DATA OLD/0/,NEW/0/,NERRS/0/,PAGE/0/
DATA LTNE/50/,MAXLIN/50/
DATA NTTYPE/5/,TYPE/1H+,1H-,1HC,1HU,1HE/,ENDSW/.FALSE./
C
C FORMAT FOR BASE AND INSERT TO TEST
1000 FORMAT(72A1,4X,I4)
C
C FORMAT FOR TEST
2000 FORMAT(72A1,2I4)
C
C FORMAT FOR CMND
5000 FORMAT(B0A1)
C
C FORMAT FOR TITLE LINE
6001 FORMAT(1H1,72A1,4X,4HPAGE,I4/)
C
C FORMAT FOR COPY ACTION (TYPE -, C, U, OR E)
6002 FORMAT(1X,A1,1X,72A1,2I4,1X,A1)
C
C FORMAT FOR CMND CARD AND INSERT
6003 FORMAT(1X,A1,1X,72A1,4X,I4,1X,A1)
C
C ERROR MESSAGE FORMATS
6101 FORMAT(6H ERR #,I4,1X,1B(1H*),20H INVALID EDIT TYPE. *36(1H*))
6102 FORMAT(6H ERR #,I4,1X,1B(1H*),26H COMMAND OUT OF SEQUENCE. *
1 30(1H*))


```

Figure 2.1 First Part of an Edit Command Deck to Create a Base Module

C....	MEDIT VERSION TWO	MEDIT VERSION TWO	MEDIT VERSION TWO	PAGE
♦ C....	MEDIT VERSION TWO	MEDIT VERSION TWO	MEDIT VERSION TWO	1 ♦
♦ C.-.	UNIT USAGE	ENVIRONMENT TYPE		2 ♦
♦ C.-.	1 BASE (SOURCE TO BE EDITED)	F(80) TAPE IN		3 ♦
♦ C-	2 TEST (RESULT OF EDITING BASE)	F(80) TAPE OUT		4 ♦
♦ C	5 CMND (EDIT COMMAND CARDS)	F(80) CARD IN		5 ♦
♦ C....	6 LTST (RESULTS OF EDIT)	FA(85) PRINT		6 ♦
♦ C.-.	FIRST CMND CARD IS USED AS A TITLE (PRINTED AT TOP OF EVERY PAGE OF LIST)			7 ♦
♦ C.-.	FIRST TWO COLUMNS OF FIRST CMND CARD IDENTIFY EDIT COMMANDS			8 ♦
♦ C	ANY CMND CARD NOT A VALID EDIT COMMAND IS INSERTED IN TEST			9 ♦
♦ C....	THIRD COLUMN OF EACH EDIT COMMAND CARD IS TYPE			10 ♦
♦ C.-.	VALID EDIT TYPES ARE-			11 ♦
♦ C.-.	+N INSERT AFTER CARD NUMBER N			12 ♦
♦ C-	-N DELETE CARD N			13 ♦
♦ C	-N,M DELETE CARDS N THROUGH M INCLUSIVE			14 ♦
♦ C....	CN MAKE CARD N A COMMENT CARD			15 ♦
♦ C.-.	CN,M MAKE CARDS N THROUGH M COMMENT CARDS			16 ♦
♦ C-	UN UNCOMMENT CARD N			17 ♦
♦ C-	UN,M UNCOMMENT CARDS N THROUGH M INCLUSIVE			18 ♦
♦ C	ENXAXRX EDIT CARD N			19 ♦
♦ C	EN,MXAXRX EDIT CARDS N THROUGH M INCLUSIVE			20 ♦
♦ C				21 ♦
♦ C				22 ♦
♦ C	SWITCH THAT INDICATES ENDFILE ON BASE			23 ♦
♦ C	LOGICAL#1 ENDSW			24 ♦
♦ C	DECLARE ALL OTHER VARIABLES TO BE TYPE INTEGER			25 ♦
♦ C	INTEGER#2NTYPE,TYPE(5)			26 ♦
♦ C	INTEGER#2CD(72),TITLE(72),OLDCD(72),MED(6)			27 ♦
♦ C	INTEGER#2OLD,NFH,PAGE,LTHE,MAXLN,NERRS,CHAR(2),THIS,NCOL			28 ♦
♦ C	INTEGER#2SPACE,COMMA,DIGIT(10)			29 ♦
♦ C	INTEGER#2 NUM,I			30 ♦
♦ C*****	FOLLOWING LOGIC IS FOR TEXT EDIT FEATURE ONLY *****			31 ♦
♦ C	INTEGER#2 NEWCD(73),OLD,ROLD,LNEW,RNEW,ICOL,JCOL,KCOL			32 ♦
♦ C*****	PRECEDING LOGIC IS FOR TEXT EDIT FEATURE ONLY *****			33 ♦
♦ C	DATA INITIALIZATION			34 ♦
♦ C	DATA SPACE/1H /*,COMMA/1H/*,MED/1H,1HE,1HO,1HI,1HT,1H /			35 ♦
♦ C	DATA DIGIT/1H,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/			36 ♦
♦ C	DATA OLD/0/,NEW/0/,NERRS/0/,PAGE/0/			37 ♦
♦ C	DATA LINE/50/,MAXLN/50/			38 ♦
♦ C	DATA KTYPE/5/,TYPE/1H*,1H-,1HC,1HL,1HE/,ENDSM//,FALSE./			39 ♦
♦ C	FORMAT FOR BASE AND INSERT TO TEST			40 ♦
♦ C	1000 FORMAT(72A1,4X,T4)			41 ♦
♦ C	FORMAT FOR TEST			42 ♦
♦ C	2000 FORMAT(72A1,2T4)			43 ♦
♦ C	FORMAT FOR CMND			44 ♦
♦ C	5000 FORMAT(80A1)			45 ♦
♦ C	FORMAT FOR TITLE LINE			46 ♦
♦ C	6001 FORMAT(1H1,72A1,4X,4HPAGE,14/)			47 ♦
♦ C	FORMAT FOR COPY ACTION (TYPE - , C, U, OR E)			48 ♦
♦ C	6002 FORMAT(1X,A1,1X,72A1,2T4,1X,A1)			49 ♦
♦ C	FORMAT FOR CMND CARD AND INSERT			50 ♦

Figure 2.2 First Page of MEDIT Output When Base Module is Created

FIGURE 2.3 - JOB CONTROL TO CREATE A BASE MODULE

PART 1 - AT USA AVSCOM, ST. LOUIS

```
//MV4L1D JOB (1T01,M999),'VALID JOB CARD'
/*SETUP   DMSENV
//NEW EXEC MEDITFLG
//EDIT.TEST DD DISP=(NEW,KEEP),DSN=YOUR.SOURCE.BASE1,
//      JINIT=2314,VOL=SER=DMSENN
//EDIT.SYSIN DD *
...
  SOURCE DECK
...
/*
//FORT.SYSLIN DD DISP=(NEW,KEEP),DSN=YOUR.BINARY.BASE1,
//      VOL=SER=DMSEVN,
//      JUNIT=2314,SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
...
  DATA
...
/*
AVAILABLE PROCS-
MEDIT - FOR MEDIT ONLY
MEDITA, MEDITAG, MEDITAL, MEDITALG - FOR ASSEMBLER-G
MEDITF, MEDITFG, MEDITFL, MEDITFLG - FOR FORTRAN-G
MEDITH, MEDITHL, MEDITHLG - FOR FORTRAN-H
MEDITP, MEDITPG, MEDITPL, MEDITPLG - FOR PL/I
MEDITWAT - FOR WATFIV

SUFIX G MEANS LOAD-AND-GO
SUFIX L MEANS LINK-EDIT
SUFIX LG MEANS LINK-EDIT AND GO
NOTE - PROCS AUTOMATICALLY USE RODMAN CAL-COMP LIBRARY.
```

PART 2 - AT USAMSSA (PENTAGON)

```
//ZXNAZERO JOB (2401,1Z07,1,20,909,1,1,P),
//      '1H VALID JOB CARD U',CLASS=0
/*ROUTE XEQ CAA-RIA
//NEW EXEC MEDITFLG
//EDIT.TEST DD DISP=(NEW,KEEP),DSN=YOUR.SOURCE.BASE1,
//      JINIT=3330,VOL=SER=JSAAWC
//EDIT.SYSIN DD *
...
  SOURCE DECK
...
/*
//FORT.SYSLIN DD DISP=(NEW,KEEP),DSN=YOUR.BINARY.BASE1,
//      VOL=SER=USAAWC,
//      JUNIT=3330,SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
...
  DATA
...
/*
//MARSDATA DD RECORD=ZG0001-76
```

AVAILABLE PROCS SAME AS AT AVSCOM.

Figure 2.3 Job Control to Create a Base Module

FIGURE 2.3 - JOB CONTROL TO CREATE A BASE MODULE (CONT)

PART 3 - AT HOPKINS (APL)

```
//AAWCXXXX JOB (6401XXXX,C,U,N),VALID JOB CARD!
//NEW EXEC MEDITFLG
//EDIT,TEST DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.SOURCE.BASE1,
//      JINIT=3330,VOL=SER=JSAAWC
//EDIT,SYSIN DD *
    ...
      SOURCE DECK
    ...
/*
//FORT,SYSLIN DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.BINARY.BASE1,
//      VOL=SER=USAAWC,
//      JINIT=3330,SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
    ...
      DATA
    ...
/*
      ...

AVAILABLE PROCS SAME AS AT AVSCOM.
```

Figure 2.3 Job Control to Create a Base Module (cont)

3.0 CHANGING AND TESTING

Once a base module has been created, changes to it may be required. The program MEDIT can be used to make these changes and produce an updated or test version of the module. This test module should be used to compile and test the changes.

The MEDIT commands to make changes to the base module are:

- a. ..+n to insert cards after card n
- b. ..-n,m to delete cards n through m inclusive
- c. ..-n same as ..-n,n
- d. ..Cn,m to place a 'C' in column 1 of cards n through m inclusive
- e. ..Cn same as ..Cn,n
- f. ..Un,m to place a space in column 1 of cards n through m inclusive
- g. ..Un same as ..Un,n
- h. ..En,m!old!new! to edit test on cards n through m inclusive
(See below)
- i. ..En!old!new! same as ..En,n!old!new!

Any card that is not an MEDIT command is treated as a card to be inserted. All card numbers refer to the sequence numbers in the base module and must appear in numerical order. Cards may be inserted in place of deleted cards by coding the appropriate ..- card followed by the card(s) to be inserted. Cards can be inserted before the first card in the deck by placing them immediately after a ..+0 card at the beginning of the edit command deck. A sample to illustrate all this is given in Figure 3.1. MEDIT output from an update run is shown in Figure 3.2.

A short summary of the logic of MEDIT will serve as a detailed description of the various MEDIT commands. In the main loop of MEDIT, an input card is read. If this card is not an MEDIT command (columns 1 and 2 are tested), then this card is written to the test module. If the input card is an MEDIT command, then the appropriate action is taken as follows.

If the input card is an "insert" command (..+n), then the base module is copied to the test module with the base module card numbered n being the last card copied. In conjunction with the treatment of non-MEDIT

```
••+0      CONVERT MEDIT VERSION 2 TO CDC-6000 SCOPE
        PROGRAM MEDIT(RASF,TTEST,INPUT,OUTPUT,
        1      TAPE1=BASE,TAPE2=TEST,TAPE5=INPUT,TAPE6=OUTPUT)
••-24
        LOGICAL END$W
••E26.32!INTEGER*2!INTEGER!
••E33!50!55!
••+39
C
••-65
        READ(5,5000) CHND
        IF(FOF(5).NE.0) GO TO 900
••-88
        100 READ(5,5000) CHND
        IF(FOF(5).NE.0) GO TO 900
••-24A
        READ(1,1000) OLDCD,OLD
        IF(FOF(1).NE.0) GO TO 192
••-283,284
        401 READ(1,1000) OLDCD,OLD
        IF(FOF(1).EQ.0) GO TO 400
```

Figure 3.1 Sample Edit Command Deck

..MEDIT CONVERT MEDIT VERSION 2 TO CDC-6000 SCOPE		PAGE	1
..+0	CONVERT MEDIT VERSION 2 TO CDC-6000 SCOPE		
♦	PROGRAM MEDIT(BASE,TEST,INPUT,OUTPUT,		1 ♦
♦	1 TAPE1=BASE,TAPE2=TEST,TAPE5=INPUT,TAPE6=OLTPUT)		2 ♦
..-24			
-	LOGICAL#1 END\$H	24	25 -
♦	LOGICAL END\$H	26	26 ♦
..E26,32!INTEGER*2!INTEGER !			
E	INTEGER "TYPE",TYPE(5)	26	28 E
E	INTEGER CHND(72),TITLE(72),OLDCD(72),MED(6)	27	29 F
E	INTEGER OLDCD,FH4,PAGE,I,TITLE,MAXLIN,NERRS,CHAR(2),THIS,NCOL	28	30 E
E	INTEGER SPACE,COMMA,DIGIT(10)	29	31 F
E	INTEGER NUM,I	30	32 F
E	***** FOLLOWING LOGIC IS FOR TEXT EDITT FEATURE ONLY *****	31	33 F
E	INTEGER NENCD(73),OLD,ROLD,LNEW,RNEW,ICOL,JCOL,KCOL	32	34 F
..ERR150!55!			
E	DATA LINF/55/,MAXLIN/55/	38	40 E
..+39			
♦ C		42	+
..-65			
-	READ(5,5000,END=900) CHND	65	67 -
♦	READ(5,5000) CHND	66	68 +
♦	IF (EOF(5),NE,0) GO TO 900	67	69 +
..-PA			
-	100 READ(5,5000,END=900) CHND	88	91 -
♦	100 READ(5,5000) CHND	89	92 +
♦	IF (EOF(5),NE,0) GO TO 900	90	93 +
..-248			
-	READ(1,1000,END=192) OLDCD,OLD	248	252 -
♦	READ(1,1000) OLDCD,OLD	249	253 +
♦	IF (EOF(1),NE,0) GO TO 192	250	254 +
..-283,284			
-	401 READ(1,1000,END=402) OLDCD,OLD	283	288 -
-	GO TO 400	284	288 -
♦	401 READ(1,1000) OLDCD,OLD	285	289 +
♦	IF (EOF(1),EQ,0) GO TO 400	286	290 +

NO ERRORS ENCOUNTERED. NORMAL END TO MEDIT.

Figure 3.2 MEDIT Output from Sample Edit Command Deck

commands, this will cause cards to be inserted in the appropriate position in the test module.

If the input card has any of the other forms (..-n or ..-n,m or ..Cn or ..Cn,m or ..Un or ..Un,m or ..En or ..En,m), then the base module is copied to the test module with the base module card numbered n-1 being the last card copied. This positions both the base and test modules for the appropriate action.

If the input card is a "delete" command (..-n or ..-n,m), then the base module is read until the appropriate base module card numbered m has been read. Since these cards are not copied to the test module, they are deleted from the test module. In conjunction with the treatment of non-MEDIT commands, this can be used to insert cards in place of the deleted cards.

If the input card is a "comment" command (..Cn or ..Cn,m), then the base module is copied to the test module with the character C replacing the first column of each card. This converts each of the cards to a FORTRAN comment card in the test module.

If the input card is an "uncomment" command (..Un or ..Un,m), then the base module is copied to the test module with the blank character replacing the first column of each card. This converts each of the cards to a FORTRAN statement in the test module. The "uncomment" command should be used only on cards that have been "commented" when the base module was created.

If the input card is an "edit text" command (..En or ..En,m), then the first character after the line number on the input card is used as a character string delimiter to determine the "old string" and "new string" on the input card. In the example "edit text" command:

..E54,59!THIS!THOSE!

The exclamation point (!) is the delimiter, the "old string" is the characters THIS and the "new string" is the characters THOSE. Any character except a comma may be used as a delimiter. The delimiter cannot be used in either the "old string" or the "new string."

The basic operation of the "edit text" operation is to copy cards from the base module to the test module. However, before each card is copied, it is searched for occurrences of the "old string." Every occurrence

of the "old string" is replaced by the "new string" before the card is copied. Thus, in the example "edit text" command, every occurrence of the characters THIS on cards 54 through 59 will be replaced by the characters THOSE in the test module.

Making changes to a base module and testing these changes proceeds as follows:

a. An edit command deck is used to make changes to the base module producing a test module. Any old binary decks for this module should be purged.

b. This test module is compiled and tested. The resulting binary deck may be saved but the test module is not.

c. If this test module is still not correct, additional changes are added to the edit command deck and the process repeated from step a. above.

Note that the test module is not saved, only the compiled version of the test module. Any new changes will be made to the base module so that the edit command deck will serve as a detailed record of every change. Returning to the base module for every change also prevents a proliferation of versions of the program and the attendant confusion and waste of computer storage (disk or tape).

The test module which is produced by a MEDIT run has two sets of sequence numbers. One set, in columns 77 - 80, are sequential starting with 1. The other set, in columns 73 - 76, are copied from the base module. For cards that have been inserted, columns 73 - 76 are blank. This set of sequence numbers are copied from the base module to facilitate making additional changes since all edit commands refer to the sequence numbers in the base module. The listing of the test module can be used to locate additional changes to the base module and the corresponding edit command cards placed appropriately in the edit command deck.

The various parts of Figure 3.3 show the job control to make changes to a library at each installation where MEDIT has been implemented. Please note that the module of the changed source is not saved. These changes should be made for every test run until a new base module is created (Chapter 4). A new base module should not be created until all potential changes have been proven correct.

FIGURE 3.3 - JOB CONTROL TO EDIT A MODULE

PART 1 - AT USA AVSCOM, ST. LOUIS

```
//MV4LID JOB (1T01,M999),'A VALID JOB CARD'
//EDIT EXEC MEDITFLG
//EDIT.BASE DD DISP=OLD,DSN=YOUR.SOURCE.BASE1,
//    UNIT=2314,VOL=SER=DMSEVN
//EDIT.SYSIN DD *
..+0      TITLE CARD - APPEARS AT TOP OF EVERY PAGE
...
    EDIT COMMAND DECK
...
/*
//EDIT.PURGE DD DISP=(MOD,DELETE),DSN=YOUR.BINARY.BASE1,
//    UNIT=2314,VOL=SER=DMSEVN --- TO PURGE OLD BINARY
//FORT.SYSLIN DD DISP=(NEW,KEEP),DSN=YOUR.BINARY.BASE1,
//    UNIT=2314,VOL=SER=DMSEVN,
//    SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
...
    DATA
...
/*
```

PROCS SAME AS AT AVSCOM. PROCS LISTED IN FIG. 2.3.

PART 2 - AT USAMSSA (PENTAGON)

```
//ZXVAMERO JOB (2401,1Z07,1,20,999,1,1,P),
//    '1H VALID JOB CARD U',CLASS=0
//ROJTE XEQ CAA-RIA
//EDIT EXEC MEDITFLG
//EDIT.BASE DD DISP=OLD,DSN=AAWC.YOUR.SOURCE.BASE1
//EDIT.SYSIN DD *
..+0      TITLE CARD - APPEARS AT TOP OF EVERY PAGE
...
    EDIT COMMAND DECK
...
/*
//EDIT.PURGE DD DSN=AAWC.YOUR.RINARY.BASE1,
//    DISP=(MOD,DELFT) --- TO DELETE OLD BINARY
//FORT.SYSLIN DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.BINARY.BASE1,
//    UNIT=3330,VOL=SER=USAAC,
//    SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
...
    DATA
...
/*
```

PROCS SAME AS AT AVSCOM. PROCS LISTED IN FIG. 2.3.

Figure 3.3 Job Control to Edit a Module

FIGURE 3.3 - JOB CONTROL TO EDIT A MODULE (CONT)

PART 3 - AT HOPKINS (APL)

```
//AAWCNAME JOB-(6401,USER,C,U,N),'VALID JOB CARD'
//EDIT EXEC MEDITFLG
//EDIT.BASE DD DISP=OLD,DSN=AAWC.YOUR.SOURCE.BASE1
//EDIT.SYSIN DD *
..+0      TITLE CARD - APPEARS AT TOP OF EVERY PAGE
...
EDIT COMMAND DECK
...
/*
//EDIT.PURGE DD DSN=AAWC.YOUR.BINARY.BASE1,
//    DISP=(OLD,DELETE) --- TO PURGE OLD BINARY
//FORT.SYSLIN DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.BINARY.BASE1,
//    UNIT=3330,VOL=SER=USAACW,
//    SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
...
DATA
...
/*
PROCS SAME AS AT AVSCOM. PROCS LISTED IN FIG. 2.3.
```

Figure 3.3 Job Control to Edit a Module (cont)

4.0 CREATING A NEW BASE MODULE

After changes have been tested and proven, these changes can be made permanent by creating a base module that reflects all these changes. This is accomplished by changing the job control to save the test module and using the same MEDIT command deck that resulted in the proven test module.

The changes to the old base module are transferred to the new base module when the test module is saved. The new base module is renumbered with the new sequence numbers in columns 77 - 80.

The cost of an edit and compile run is determined by the length of the base module and not by the number of changes being made. Therefore, the creation of a new base module should be avoided whenever possible.

The time to create a new base module is either:

a. When production runs of the test version are nearly completed and the user is very sure that no additional changes will be required; or

b. When the edit command deck exceeds 1000 cards, approximately one-half a box of cards.

When a new base module is created, the listing and binary (compiled) deck should also be kept as a reference for as long as this base module is kept. After the new base module has been successfully created and compiled, the old base module should be purged together with its listing and binary (compiled) deck.

The various parts of Figure 4.1 give the job control for updating a library at the various installations where MEDIT has been implemented.

FIGURE 4.1 - JOB CONTROL TO UPDATE BASE MODULE.

PART 1 - AT USA AVSCOM, ST. LOUIS

```
//MV4LIU JOB (1T01,M999),'A VALID JOB CARD'
//SETUP DMSENN
//EDIT EXEC MEDITFLG
//EDIT.TEST DD DISP=(NEW,KEEP),DSN=YOUR.SOURCE.BASE2,
//    UNIT=2314,VOL=SER=DMSENN,SPACE=(TRK,(1,1),RLSE)
//EDIT.BASE DD DISP=OLD,DSN=YOUR.SOURCE.BASE1,
//    UNIT=2314,VOL=SER=DMSENN
//EDIT.SYSIN DD *
..+0      TITLE CARD - APPEARS AT TOP OF EVERY PAGE
...
EDIT COMMAND DECK
...
/*
//EDIT.PURGE DD DISP=(MOD,DELETE),DSN=YOUR.BINARY.BASE1,
//    UNIT=2314,VOL=SER=DMSENN --- TO PURGE OLD BINARY
//FORT.SYSLIN DD DISP=(NEW,KEEP),DSN=YOUR.BINARY.BASE2,
//    UNIT=2314,VOL=SER=DMSENN,
//    SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
...
DATA
...
/*

```

PROCS SAME AS AT AVSCOM. PROCS LISTED IN FIG. 2.3.

PART 2 - AT USAMSSA (PENTAGON)

```
//ZX4AMERO JOB (2401,1Z07,1,20,999,1,1,P),
//    '1H VALID JOB CARD U',CLASS=0
//ROJTE XEQ CAA-RIA
//EDIT EXEC MEDITFLG
//EDIT.TEST DD DISP=(NEW,CATLG),DSN=A4WC.YOUR.SOURCE.BASE2,
//    UNIT=3330,VOL=SER=USA4WC,SPACE=(TRK,(1,1),RLSE)
//EDIT.BASE DD DISP=OLD,DSN=A4WC.YOUR.SOURCE.BASE1
//EDIT.SYSIN DD *
..+0      TITLE CARD - APPEARS AT TOP OF EVERY PAGE
...
EDIT COMMAND DECK
...
/*
//EDIT.PURGE DD DSN=A4WC.YOUR.BINARY.BASE1,
//    DISP=(MOD,DELETE) --- TO DELETE OLD BINARY
//FORT.SYSLIN DD DISP=(NEW,CATLG),DSN=A4WC.YOUR.BINARY.BASE2,
//    UNIT=3330,VOL=SER=USA4WC,
//    SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
...
DATA
...
/*

```

PROCS SAME AS AT AVSCOM. PROCS LISTED IN FIG. 2.3.

Figure 4.1 Job Control to Update Base Module

FIGURE 4.1 - JOB CONTROL TO UPDATE BASE MODULE (CONT).

PART 3 - AT HOPKINS (APL)

```
//AAWCNAME J0R (6401,USER,C,U,N),'VALID JOB CARD'
//EDIT EXEC MEDITFLG
//EDIT TEST DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.SOURCE.BASE2,
//    UNIT=3330,VOL=SER=USAACW,SPACE=(TRK,(1,1),RLSE)
//EDIT BASE DD DISP=OLD,DSN=AAWC.YOUR.SOURCE.BASE1
//EDIT SYSIN DD *
..+0      TITLE CARD - APPEARS AT TOP OF EVERY PAGE
...
    EDIT COMMAND DECK
...
/*
//EDIT PURGE DD DSN=AAWC.YOUR.BINARY.BASE1,
//    DISP=(OLD,DELETE) --- TO PURGE OLD BINARY
//FORT.SYSLIN DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.BINARY.BASE2,
//    UNIT=3330,VOL=SER=USAACW,
//    SPACE=(TRK,(1,1),PLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
...
    DATA
...
/*
PROCS SAME AS AT AVSCOM. PROCS LISTED IN FIG. 2.3.
```

Figure 4.1 Job Control to Update Base Module (cont)

A.0 APPENDIX - LISTING OF THE MEDIT PROGRAM

NOTE: This is a listing of the CDC-6000 version of MEDIT. This source was the result of the editing described in Figures 3.1 and 3.2.

PAGE
 04/06/76 12:03:21.
 FTN 4.3+P93
 7474 CPT#2
 04/06/76 12:03:21.
 PROGRAM WFCIT(ASF,TEST,INPUT,OUTPUT)
 1 1 TAF1=ASF,TAF2=TEST,TAF3=INPUT,TAPF=OUTPUT
 2 2 UNIT 14(4)
 3 3 WERIT VERSION TWO RECIT WSRST TWO EDIT VERSION TWO
 4 4 ENVIRONMENT TYPE
 5 5 F(AN) TAPE IN
 6 6 F(R) TAPE OUT
 7 7 F(W) CARD IN
 8 8 F(HS) PRINT
 9 9
 10 10 FIRST CARD IS USED AS A TITLE
 (PRINTED AT TOP OF EVERY PAGE OF LIST)
 11 11 FIRST TWO COLUMNS OF FIRST CARD IDENTIFY EDIT COMMANDS
 12 12 ANY CARD CONTAINING A VALID EDIT COMMAND IS INSERTED IN TEST
 13 13 THIRD COLUMN OF EACH EDIT COMMAND CARD IS TYPE
 14 14 VALID EDIT TYPES ARE:
 15 15
 * AN INSERT AFTER CARD NUMBER N
 * DELETED CARD N
 * DELETED CARDS N THROUGH M INCLUSIVE
 16 16
 * MOVE CARD N A COMMENT CARD
 17 17 MOVE CARDS N THROUGH M COMMENT CARDS
 18 18
 * INCREMENT CARD N
 19 19 INCREMENT CARDS N THROUGH M INCLUSIVE
 20 20 ENFORCE EDIT CARD N
 21 21 ENFORCE EDIT CARDS N THROUGH M INCLUSIVE
 22 22
 23 23
 24 24
 25 25 SWITC THAT INDICATES ENFILE ON BASEF
 26 26
 27 27 LOGICAL ENFILE
 28 28 DECLARE ALL OTHER VARIABLES TO BE TYPE INTEGER
 29 29
 30 30
 31 31
 32 32
 33 33
 34 34
 35 35
 36 36
 37 37
 38 38
 39 39
 40 40
 41 41
 42 42
 43 43
 44 44
 45 45
 46 46
 47 47
 48 48
 49 49
 50 50
 51 51
 52 52
 53 53
 54 54
 55 55

PROGNAME: EREDIT
 74 / 74 C01=2
 FTN 4.3.P393
 04/06/76 12.03.21.
 PAGE 2

```

1 JN(1)(*)  

6141 FOPEN T4M TDR #14,1X,181H*) *8H COLUMN(.12,2H)*.A1,30H DOES NOT 0 55 58
1FLNIT(0,0) SH14C .131H*)  

6142 FDLVTR T4M TDR #14,1X,1H(*)*7,29H OLD STRING HAS ZERO LENGTH. +27( 57 60
1143)  

6143 FDLVTR T4M TDR #14,1X,181H*) *8H COLUMN(.12,2H)*.A1,30H DOES NOT 0 59 62
1FLNIT(0,0) SH14C .131H*)  

6150 FDLVTR T4M TDR #14,1X,1H(*)*7,40H NON-BLANK DATA LOST AS RESULT OF 60 63
1 ENTR. 151H*)  

C KEN1 INST CMC-C CARD  

MEAN(S,SOUD) COUNT  

IF(EF(1),NE,0) GO TO 900 62 65
70 C COPY TC TITLE  

DO 1 NCOL=1,72 64 68
1 TITLE(FCOL,1,NCOL)  

C TS FLIST CWD CARD A VALIN EDIT COMMAND 65 69
DO 2 THISLTYPE  

IF(ITR,THISLTYPE,FN,CMD(3)) GO TO 3 69 73
2 CONTINUE  

C FIRST CARD CARD NOT A VALID EDIT COMMAND 70 74
C SUFFIX CMD WITH AN NN# CMD WHERE N IS A NULL CHARACTER 71 75
CMD(1)=U  

CMD(2)=U  

GO TO 101 72 76
C EREDIT CARD IS A VALID EDIT COMMAND  

C SET CMD TO IDENTIFY Future EDIT COMMANDS 73 77
3 CHAR(1) SECOND(1)  

CHAR(2) SECOND(2)  

C ALTN TITLE TO USE NAME OF MEDIT 74 78
DO 4 NCAL=1,6 75 79
4 TITLE(FCOL*2,FCOL)  

GO TO 103 76 80
C START OF MAIN EDIT LOOP 77 81
C OPEN MEAT CARD CARD 78 83
100 READ(S,SOUD) CHNG 79 83
TF(FR(1),NE,0) GO TO 500 80 84
C THIS IS AN EDIT COMMAND 81 85
TF(CMD(1)*2,CHAR(1)*AND, CMD(2)*EQ,CHAR(2)) GO TO 103 82 86
C NO - INSERT THIS CARD 83 87
101 ASSIN 102 FC INIT 84 88
C 102 FC INIT 85 89
102 NE=2,FW1 86 90
NTRT(14,6,0) TYPE(1)*CMD*NEW,TYPE(1) 87 91
GTRT(2,1,0) CMD*IF,W 88 92
C THIS CMD CARD IS AN EDIT COMMAND 89 94
103 ASSIN 104 FC INIT 90 95
C 103 ASSIN 105 91 96
104 ASSIN 106 FC INIT 92 97
105 107 93 98
106 108 94 99
107 109 95 100
108 110 96 101
109 111 97 102
110 112 98 103
111 113 99 104
112 114 100 105
113 115 101 106
114 116 102 107
C WHAT TYPE IS IT 103 108
DO 105 THISLTYPE 104 109
IF(CMD(3)*EC,TYPE(TM19)) GO TO 106 105 110
C CONTINUE 106 111
NEDIT*4*1 107 112
WHITE(*EOLU) NEWRS 108 113
109 114

```

PROGRAM EDITT 74/74 001*2 PAGE 3
 FTN 4.3+P393 04/06/76 12:03:21.
 LIN#17*E*
 GO TC 101
 C COMMAND CMC IS A VALID EDIT COMMAND
 C EVALUATE FIRST NUMBER
 106 ASSIGN 107 TC EXIT
 NCOL=3
 GO TC 100
 C IS COMMAND IN SEQUENCE
 C 107 IF(Y=.6-.CLC) GC TO 103
 C NO SIGNAL FOR AND RESET NUM
 AFRESET(NCOL)*
 WRITE(*,OLU2) AF44S
 LINE#17*E*
 GO TC 100
 C COPY FROM HASF TO TEST
 108 ASSIGN LUCN TC EXIT
 C SET HUP IF THIS IS AN INSERT COMMAND.
 IF(TIS&TC.1) NUMBER=1
 GO TC 400
 C TIS THIS COMMAND AN INSERT
 109 IF(TIS.E.1) GO TO 104
 C PNT CCMAN IS NOT INSERT
 C TS TRIM A SPECIFIC NUMBER
 110 IF(CS.FNCCL)*.CCMVA) GO TO 111
 C YFC. EVALUATE IT
 ASSIGN 111 TC EXIT
 GO TC 310
 C LOOP TO ALTFN BASE
 C SFT EXIT FWD LINE COUNT ROUTINE
 111 ASSIGN 150 TC EXIT
 IF(TIS.E.2) ASSIGN 151 TC EXIT
 ***** FOLLOWING LOGIC IS FOR TEXT EDIT FEATURE ONLY
 C SET AFACD SWITCH IN CASE NOT AN EDIT COMMAND.
 C KEYCD(1)BHF
 C AFACD SWITCH IN CASE NOT AN EDIT FEATURE ONLY
 145 *****
 C TIS TYPE AN EDIT
 IF(TIS.E.5) GO TO 112
 C YFS. GET CMC AND SET STRING LIMITS
 LONGINTL.1
 GO TO 112 PULLBLD.1,72
 IF(CS.FNCCL)*.CPNU(NCOL)) GO TO 143
 C INVALID DELIMITER. SIGNAL ERROR AND IGNORE CMD
 AFRESET(NCOL) AFHKS*NCOL.CPNCD(NCOL)
 LINE#17*E*
 GO TC 100
 C 143 IF(TIS.E.6)LCCL) GC TO 144
 C NO SIGNAL FOR AND IGNORE
 AFRESET(NCOL)*
 WRITE(*,OLU2) AF44S
 LINE#17*E*
 GO TC 100
 C 144 LRF=OLU1
 GO DFLCCL.1
 NO 145 RHEAFLR=72
 IF(CMC(FNCCL).FO.CPMN(NCOL)) GC TO 146
 145

PAGE
 04/06/76 12:03:21.
 FTN 4.3#P34
 04/06/76 12:03:21.
 PROGRAM #PNT 7674 CPT#2
 145 CONT'D
 C TYPICAL EDITTING
 C NERCODE(*1)
 C WRITE(*COL+3) AFERS.NCOL+CND(NCOL)
 C LINE FEED +1
 GO TO 100
 100 REN IDENTICAL
 C ***** EDITTING LOGIC IS FOR TEXT EDIT FEATURE ONLY
 C APE OF UC/IF
 C 112 IF(UC.EQ.' ') GO TO 100
 C 100 PLIF(K, ACTION PER THIS
 GO TO (100,200,120,130,140),THIS
 100 UCDC(I)=TAC(I) A COMMENT
 GO TO 200
 C 101 IDENT CLPDC
 C 103 UCDC(I)=SPACE
 GO TO 200
 C EXIT TEST
 C ***** EDITTING LOGIC IS FOR TEXT EDIT FEATURE ONLY
 C FMT(UCDC), PUTTING RESULTS INTO NEWCD
 C 104 IDENTICAL UCDC SWITCH AND COLUMN COUNTS
 104 DO 102 151*173
 102 UCDC(I)=SPACE
 ICLEN
 KCOL=N
 C TC(I,NL) STRING IN OLDCC STARTING AT NCOL+1
 DO 153 PCOL=OLDPCOL
 JCOL=JCOL+1
 TF(LNL,LF,72) GC TO 152
 C COMPARE RETURN OLDCC - USE SPACE
 IF(SPACE.EQ.CAN(KCOL)) GO TO 157
 GO TO 153
 C PRINTS THE 4C CHARACTERS
 152 IF(UCDC(I).NE.CAN(KCOL)) GC TO 157
 151 CONTINUE
 C FIRST STRING STARTS AT CCOLUMN NCOL+1
 C COPY 4C STRING TO NEWCD (IF NOT A NULL STRING)
 IF(FIRST,LF,72) GC TO 154
 ON 155 PCOL=LNL+1+4
 ICOL=ICOL+1
 IF(LNL,LF,72) GC TO 154
 C IF(FIRST,LF,72).EQ. SPACE) NEWCD(I3)=CND(NCOL)
 GO TO 155
 C MOVE ONE CHARACTER FROM NEW STRING
 156 MF=CND(I3)=CND(NCOL)
 155 CONTINUE
 C SFT NCOL TO LAST COLUMN OF OLD STRING
 158 NCOL=LNL
 GO TO 159
 C FIRST STRING DOES NOT START AT NCOL+1
 157 NCOL=COL+1
 ICOL=ICOL+1
 IF(LNL,LF,72) GC TO 158
 C FILE SYNC NCOL = SET SWITCH

PROGRAM #F1IT 74/74 OPT#2
 F1N 4.3#393 06/06/76 12.03.21.
 PAGE 5

```

210      IF (I.F(171).EQ. SPACE) NEWCD(73)=OLDCD(NCOL)
          GO TO 159
          C MOVE ONE CHARACTER FROM OLD_CD
          C NEW_CD(I+1)=CD(I+1)
          C AND "I" IS WITH OLD_CD
          159 IF (I.EQ. LT.72) GO TO 151
          C WRITE(I,E) TO CLICL
          DO 161 ICL=1,72
          161 CLICL=LENCD(ICL)
          GO TO 203
          C***** EFFECTUATE LOGIC IS FOR TEXT EDIT FEATURE ONLY *****
          C IF NOT DELETED, COPY OLD_CD TO TSFT
          190N NEE=N.F.+1
          IF I.GT. NEE .1. COPY OLD_CD TO TSFT
          191 WRITE(I,200) CLFCC,NEW,NF
          C 191 WRITE A RESULT OF EDIT
          C TSFT IS A SUBROUTINE TYPE (THIS).OLDCD.DLD+NEW+TYPE (THIS)
          C 192 TSFT OR
          IF (NEC(I).LT.1) EC. SPACE) GO TO 160
          C 'N' - DATA LOST
          NEE=N.EE+1
          WRITE(I,200) AFRS
          LINE(I)=LINE(I)+AFRS
          160 CONTINUE
          C IF(I.GT.1) READ C,40 FROM BASE AND LCOP
          READ(I,100) OLCD(1),OLCD(0)
          IF (I.F(1).NE.0) GO TO 192
          GO TO 112
          C ENDFILE ENCOUNTERED ON BASE
          192 ERNO=-1
          GO TO 196
          C SUBROUTINE TO COUNT LINES AND PRINT TITLE AT TOP OF EACH PAGE
          200 LINE(I)=1
          IF (I.TP.EQ.'MAINT') GO TO 201
          C COUNT TITLE
          C PAGE=240,I
          WRITE(4,901) TITLE,PAGE
          L1:F=]
          201 CONTINUE
          GO TO 196
          C 300 NUMBER
          300 NUMBER=1
          C SUBROUTINE TO EVALUATE NUM
          301 NUMBER(I)=1
          C WHAT DIGIT IS CMDO(NCOL)
          DO 312 I=1,10
          IF (CMDO(NCOL).EQ.DIGIT(I)) GO TO 303
          302 CONTINUE
          C CHARACTER "CT 4 DIGIT. RETURN.
          GO TO TAITS(117,11)
          C IPRINT VALUE OF NUM
          303 NUMBER(I)=I-1
          GO TO 261
          C SUBROUTINE TO COPY UP TO BASE(NUM)
          400 IF (I.EQ. 1) THEN
          C ANY COPY NEEDED
          C HAS FIRST RASF CARD BEEN READ
          C IF (OLD.LE.0) GO TO 401
          C YES. COPY BASE CARD TO TEST
  
```

PROGRAM IDENT 74/74 CPT=2

FTN 4.3+P393 04/06/76 12:03:21. PAGE 6

```

NF*+F*+
        WRITE(2,200) CLRCC,OLDANE
        OFIN,CAHC,FOR,HSTE
C 401 READ(L1,0,0) ORG(0) GO TO 400
        IF(FEFL1,0,0) GO TO 400
C 402 END=TC.
        PROFILE ENCOUNTERED ON BASE
        403 GO TO XFILE(L1,0,0)
C      PROFILE FOUND ON CPOD
        COPY TC EDC OF HSTE
C 404 NIPATIONAL
        ASSOC=4J1 TC EXIT
        GO TO 405
C 405 IF(FRMS,PL,0) GO TO 902
        IF(FRMS,PL,0) GO TO 902
C 406 YFC. LIST NUMBER OF ERRORS
        WRITE(6,901) NFRS
        STOP 14
A901 FORMAT(1X,15*2D,F4.2) FNCOUNTERFD. *57(1M*) )
C 407 WRITE(6,0)2
        STOP
C 408 NFRS=0
A902 FORMAT(45LNC ERRORS ENCOUNTERED. NORMAL END TO MEDIT.)
        EPN
        STOP (45LNC ERRORS ENCOUNTERED. NORMAL END TO MEDIT.)
        303 309

```

SYNTHETIC REFERENCE MAP (N=3)

ENTRY POINTS	OFF LINE	PREFERENCES				
10217 WPOINT						
VARIABLES	SR	TYPE				
11522 CHAD	T-TEGFR	T-TEGFR				
11144 CWD	T-TEGFR	T-TEGFR				
10641 CROWD	F-TEGFR					
11524 CIRCT	I-TEGFR					
10653 ERNS	I-TEGFR					
11146 FCOL	I-TEGFR					
11153 IFCOL	I-TEGFR					
11154 FXTT	T-TEGFR					
11154 JCNL	T-TEGFR					
11155 KCNL	T-TEGFR					
1066n LINE	T-TEGFR					
11151 LAFS	I-TEGFR					
11157 LND	I-TEGFR					
10651 PAPLN	T-TEGFR					

PROGRAM UNIT	ON	TYPE	7474	CNT=2	04/06/76	12.03.21.	PAGE
VARIABLES	ON	TYPE	45RAY	LOCATION			
11514 REN	I	INTEGER					
11516 ACNL	I	CHAR					
10466 1EERS	I	INTEGER					
10465 1FV	I	INTEGER					
11516 1EVCT	I	INTEGER					
10462 1TPPF	I	INTEGER					
11145 1UM	I	INTEGER					
10464 0LD	I	INTEGER					
11606 C1CCD	I	INTEGER		ARRAY			
10467 PARC	I	INTEGER					
11142 PRIF	I	INTEGER					
11150 RND	I	INTEGER					
10462 SPACE	I	INTEGER					
11140 THTS	I	INTEGER					
11276 TTPLF	I	INTEGER		ARRAY			
11147 TYPF	I	INTEGER		ARRAY			
FILF KANFE	None						
4162 FILE	I	FILE					
6163 CPUTUT	I	FILE					
10241 TAPE1	I	TAPE	FMT				
2041 TAPE2	I	TAPE	FIT				
4162 TAPE5	I	TAPE	PEAD5				
6163 TAPEA	I	TAPE	FAT				
2041 TFSST	I	FILE					
EXTERNALS	FOR	TYPE	ARGS	REFERENCE			
STATFILE LARFS	FFAL	1	69	93	254	290	
0 1	RFF LINE	72	71				
10242 2		76	74				
0 4		64	75				
10251 106		62	67				
10245 101		97	91				
10247 102		95	97				
10275 103		105	107				
10277 104		107	105				
0 105		111	109				

PROGRAM, UNIT	STATEMENT LARFS	74/74	74/74	REF LINE	REFERENCES	74/74	74/74	REF LINE	REFERENCES	74/74	74/74	REF LINE	REFERENCES
10314 104		115		110	276								
10320 107		123		119									
10370 108		130		123									
10394 109		135		130	291								
10343 110	Inactive	134		138									
10421 111		144		150	255								
10421 112		150		142									
10446 126		145		145									
10442 136		146		146									
10446 146		146		146									
10371 147		147		155									
10460 148		148		154									
10522 154		148		162									
10513 154		148		174	170								
10551 154		148		178	171								
10517 157		151		179	234								
10464 152		152		207	202								
10466 152		152		208	205								
10522 154		152		215	214								
10513 154		152		220	212								
10551 154		152		222	211								
10517 157		157		225	204								
10517 158		158		232	207								
10522 158		158		232	207								
10567 166		166		234	223								
10567 161		161		251	230								
10522 162		162		237	236								
10570 190		190		195	194								
10573 161		161		241	241								
10584 161		161		244	244								
10584 192		192		245	267								
10584 200		200		257	254								
10545 201		201		240	94								
10547 306		206		266	261								
10572 201		201		243	121								
10461 302		302		270	279								
10461 203		203		274	272								
10564 203		203		275	273								
10513 401		401		240	94								
10513 466		466		266	261								
10522 462		462		243	242								
10522 403		403		244	242								
10521 406		406		245	242								
10521 406		406		246	242								
10522 502		502		300	293								
10522 502		502		300	297								
10567 2000	FMT	44		200									
10567 5000	FMT	46		262									
10541 5001	FMT	48		68	92								
10542 5002	FMT	50		264									
10573 6003	FMT	52		107									
10577 6101	FMT	54		100									
10704 6102	FMT	56		114									
10715 6141	FMT	57		126									
10727 6142	FMT	58		153									
10737 6143	FMT	61		165									
10751 6150	FMT	63		175									
11122 6501	FMT	65		249									
11132 6902	FMT	66		302									
11132 6902	FMT	66		306									

PROGRAM IDENT	LABEL	TYPE	74/76	CPT#?	LENGTH	PROPERTIES
10224	1	ICOL	R20W-T1	INSTACK	3H	
10224	2	TWS	71 72	INSTACK	2R	
10247	4	KCOL	74 76	INSTACK	2R	EXITS
10304	105	TWS	87 94	INSTACK	2R	
10307	142	SCIN	105 111	INSTACK	2R	
10307	145	QEIN	152 155	INSTACK	2R	
10444	162	I	170 172	INSTACK	2R	
10440	153	KCOL	194 195	INSTACK	2H	
10474	155	KCOL	200 210	INSTACK	7A	
10525	161	ICOL	212 220	CPT	10H	
10575	302	I	236 237	INSTACK	3H	
			272 274	INSTACK	2R	
				INSTACK		EXITS
STATISTICS						
PER-SIM LENGTH			14444	H04		
DIFFER LENGTH			102354	4279		

PAGE 9

RTN 4.3.0393

04/06/76

12.03.21.

A. Dept. of the Defense

Defense Communication Engineering Office
ATTN: Mr. Roy Rosner (1)
Reston, VA

Foreign Science and Technology Center
ATTN: Mr. Reg Barta (1)
ATTN: Mr. David Howell (1)
ATTN: Mr. Thomas Pruden (1)
Charlottesville, VA

Defense Documentation Center
ATTN: TIPCR (12)
Cameron Station
Alexandria, VA 22314

B. Dept. of the Army

Commander
US Army Electronic Command
Night Vision Laboratory
ATTN: Mr. John Dehne (1)
ATTN: Mr. R. A. Oswalt (1)
Ft. Monmouth, NJ 07703

Commander
US Army Materiel and Mechanics Research Center
ATTN: AMXMR-S, Mr. R. J. Geromini (1)
ATTN: AMXMR-S, Mr. D. M. Gracia (1)
Watertown, MA 02172

Commander
US Army Ames Research Center
Air Mobility Research & Development Laboratory
ATTN: SAVDL-AS, Dr. J. Hwang (1)
ATTN: SAVDL-D, Mr. Mike Kodani (1)
Moffett Field, CA 94035

Commander
US Army Test & Evaluation Command
ATTN: AMSTE, Mr. Leon Brill (1)
Aberdeen Proving Ground, MD 21005

Commander
US Army Electronic Command
ATTN: AMSEL-SA-L, Mr. Richard Caccamise (1)
ATTN: AMSEL-IO, Mr. Henry Chambers (1)
ATTN: AMSEL-IO, Mr. Thomas Dames (2)
ATTN: AMSEL-IO, Mr. W. R. King (1)
ATTN: AMSEL-IO, Mr. E. Oltarzewski (1)
Ft. Monmouth, NJ 07703

Commander
US Army Missile Command
ATTN: AMCM, Mr. Jim Collins (2)
ATTN: AMCM, Mr. James Williams (1)
Redstone Arsenal, AL 35809

US Army Aberdeen Proving Ground
ATTN: Mr. J. Whallon (1)
Aberdeen Proving Ground, MD 21005

Commander
Picatinny Arsenal
ATTN: SARPA, Mr. A.G. Edwards (1)
ATTN: SARPA-MIS, Mr. David Grobstein (2)
Dover, NJ 07801

Commander
Edgewood Arsenal
ATTN: SAREA, Mr. S. Goldberg (2)
Edgewood, MD 21005

Commander
US Army Aviation Systems Command
ATTN: Mr. Paul Lascala (2)
ATTN: Dr. I. Peterson (1)
St. Louis, MO 63166

Commander
Watervliet Arsenal
ATTN: SARWV, Mr. James Pascale (2)
Watervliet, NY 12189

Commander
Yuma Proving Ground
ATTN: Mr. Fred Jones (1)
Yuma, AZ 85364

Director
US Army Mobility Research & Development Center
ATTN: STSFB, Mr. H. Bero (1)
ATTN: STSFB, Mr. Sam McCutchen (2)
Fort Belvoir, VA 22060

Commander

US Army Harry Diamond Laboratories
ATTN: AMXDO, Mr. Hubert Matthews (1)
ATTN: AMXDO, Ms. L. Jean Marroletti (1)
Washington, DC 20438

Director

US Army Ballistic Research Laboratory
ATTN: Mr. Sam Taylor (1)
Aberdeen, MD 21005

Commander

US Army Materiel Development and Readiness Command
ATTN: DRCMS, Mr. John Cianflone (1)
ATTN: DRCMS, Mr. John Gilbert (1)
ATTN: DRCMS, Dr. Ronald Uhlig (1)
5001 Eisenhower Avenue
Alexandria, VA 22333

Commander

US Army Armament Command
ATTN: DRSAR-SA, Mr. Stu Olson (1)
ATTN: DRSAR-MSE (1)
ATTN: DRSAR-MST (1)
ATTN: DRSAR-JCAP-E (1)
ATTN: DRSAR-JCAP-M (1)
Rock Island, IL 61201

Commander

Rock Island Arsenal
ATTN: SARRI-L (1)
ATTN: SARRI-LPL (2)
ATTN: SARRI-LA (5)
ATTN: SARRI-LE (5)
ATTN: SARRI-LR (10)
ATTN: SARRI-LS (5)
ATTN: SARRI-LW (5)
ATTN: SARRI-LR-S, Dr. Hurt (10)
ATTN: SARRI-R (1)
Rock Island, IL 61201

Commander

White Sands Missile Range
ATTN: Mr. James Field (1)
White Sands Missile Range, NM 88002

DISTRIBUTION LIST UPDATE

- - - FOR YOUR CONVENIENCE - - -

Government regulations require the maintenance of up-to-date distribution lists for technical reports. This form is provided for your convenience to indicate necessary changes or corrections.

If a change in our mailing lists should be made, please check the appropriate boxes below. For changes or corrections, show old address *exactly* as it appeared on the mailing label. Fold on dotted lines, tape or staple the lower edge together, and mail.

Remove Name From List

Old Address:

Change or Correct Address

Corrected or New Address:

COMMENTS

Date: _____ Signature: _____

Technical Report # R-TR-76-010

B-3

FOLD HERE

Return Address:

OFFICIAL BUSINESS
Penalty for Private Use \$300

POSTAGE AND FEES PAID
DEPARTMENT OF THE ARMY
DOD 314



Commander
Rock Island Arsenal
Attn: SARRI-LR-S
Rock Island, Illinois 61201

FOLD HERE

AD ACCESSION NO.
Research Directorate, General Thomas J. Rodman Laboratory
Rock Island Arsenal, Rock Island, Illinois 61201
UNCLASSIFIED

MEDIT - A Program to Edit Computer Source Programs
Prepared by: James J. Hurt
Security Class.: Unclassified
Technical Report R-TB-6-010

27 Pages, Incl Figures

A portable and easily used source text editor is described. This computer utility provides a method to maintain and modify computer source text in a manner that is very easy to use. Provides a complete audit trail of changes, and can be used on a wide variety of computers, eg IBM-360-370, CDC-6000, UNIVAC-1100, et al.

DISTRIBUTION
Approved for public release; distribution unlimited.

AD ACCESSION NO.
Research Directorate, General Thomas J. Rodman Laboratory
Rock Island Arsenal, Rock Island, Illinois 61201
UNCLASSIFIED

MEDIT - A Program to Edit Computer Source Programs
Prepared by: James J. Hurt
Security Class. (of this report): Unclassified
Technical Report R-TB-6-010

27 Pages, Incl Figures

A portable and easily used source text editor is described. This computer utility provides a method to maintain and modify computer source text in a manner that is very easy to use, provides a complete audit trail of changes, and can be used on a wide variety of computers, eg IBM-360-370, CDC-6000, UNIVAC-1100, et al.

DISTRIBUTION
Approved for public release; distribution unlimited.

AD ACCESSION NO.
Research Directorate, General Thomas J. Rodman Laboratory
Rock Island Arsenal, Rock Island, Illinois 61201
UNCLASSIFIED

MEDIT - A Program to Edit Computer Source Programs
Prepared by: James J. Hurt
Security Class.: Unclassified
Technical Report R-TB-6-010

27 Pages, Incl Figures

A portable and easily used source text editor is described. This computer utility provides a method to maintain and modify computer source text in a manner that is very easy to use, provides a complete audit trail of changes, and can be used on a wide variety of computers, eg IBM-360-370, CDC-6000, UNIVAC-1100, et al.

DISTRIBUTION
Approved for public release; distribution unlimited.

AD ACCESSION NO.
Research Directorate, General Thomas J. Rodman Laboratory
Rock Island Arsenal, Rock Island, Illinois 61201
UNCLASSIFIED

MEDIT - A Program to Edit Computer Source Programs
Prepared by: James J. Hurt
Security Class. (of this report): Unclassified
Technical Report R-TB-6-010

27 Pages, Incl Figures

A portable and easily used source text editor is described. This computer utility provides a method to maintain and modify computer source text in a manner that is very easy to use, provides a complete audit trail of changes, and can be used on a wide variety of computers, eg IBM-360-370, CDC-6000, UNIVAC-1100, et al.

DISTRIBUTION
Approved for public release; distribution unlimited.